IN THE SPECIFICATION

On the Cover Page, please amend the title to read as follows:
METHOD AND APPARATUS FOR STRIP[[P]]ING DATA ONTO A PLURALITY OF DISK DRIVES

Please replace the following paragraphs with the amended paragraphs as follows:

Paragraph [0002] beginning on page 1:

The principal requirements of a video server are the abilities to store multiple video files, as well as to continually stream any one of these files to any one of a server's multiple clients. A typical large-scale server will hold several hundred video files and be capable of streaming this data to several hundred ~~simultaneous~~ clients contemporaneously. In order for the clients to view the video without interruption, and without a large buffer required at each client site, the server must output each client's stream without interruption. Further, each client must have access to any video file on the server, so that, for example, every client could view the same file simultaneously, or each client could view a different file. Generally, video servers are capable of "VCR-like" functionality that displays a video file in normal, fast-forward, or rewind mode. This functionality generates an additional requirement on the server that a user's viewing mode changes do not incur a long latency delay, such as, for example, changes from "normal mode" to "fast-forward" should occur quickly.

Paragraph [0005] beginning on page 5:

FIG. 1 illustratively depicts a disk drive array 100 having data striped in a RAID-3 format. Specifically, the top row of boxes represents each disk 102 in the array of disks (e.g., 15 disks D0 through D14). Furthermore, each box below each disk in the array of disks represents an extent of data 110_1 though 110_p (collectively extents 110). FIG. 1 illustratively shows two files, file A and file B, each 16 extents long, striped across a disk drive array consisting of 15 disks total. The disk drive array 100 is broken into 3 parity groups 104_1 through 104_3 (collectively parity groups 104) of 5 disks each, with each parity group 104 respectively consisting of 4 data disks 106_1 through 106_3 (collectively data disks 106) and 1 parity disk 108_1 through 108_3 (collectively parity disk 108). For example, ~~the~~ a first parity group 104 comprises the first four extents of file A (i.e., extents A0-A3) illustratively written onto disks D5-8, plus the parity extent (i.e., the byte-by-byte XOR of these 4 data extents) written onto disk D9. In RAID 3, all files on the server use the same sized parity groups, so that certain disks in the array contain only parity data. In FIG. 1, the disks containing only parity data are disks 4, 9, and 14.

Paragraph [0039] beginning on page 11:

To understand the extra step in parity correction used by the RAID 3+5 algorithm, it is instructional to compare the parity correction in the RAID 3 or RAID 5 formats. Specifically, in the RAID 3 or 5 formats, if the first disk in the group failed, then the first segment of that extent (e.g., segment "a" in FIG. 4) is regenerated from a = d^g^j^X (where X is on a separate parity disk in RAID 3), which requires three XOR ("^") logic operatives. Furthermore, segments "b" and "c" in the first disk are readily accessible, as they are respectively stored in the parity segments 404_1 and 404_2 of extents 1 and 2. To regenerate segments "b" and "c" of the first extent 0 110_0, b= e^h^k^Y and [[c= f^l^l^Z]] c= f^i^l^Z, three XOR logic operatives are required to regenerate each segment "b" and "c". As such, using either RAID 3 or RAID 5, a total of 9 XOR logic operatives must be performed to recover the missing data.

Paragraph [0057] beginning on page 17:

Specifically, the portion of data buffered in server memory, which is represented by the notation A+ in cell D5 and SP 16, is transferred (streamed) to the users in group A during SP 17. Similarly, the portion of data buffered in server memory, which is represented by the notation A+ in cell D6 and SP 16, is transferred (streamed) to the users in group A during SP 18. Furthermore, the portion of data buffered in server memory, which is represented by the notation A+ in cell D7 and SP 16, is transferred (streamed) to the users in group A during SP [[18]] 19. During the service period SP 20, the server 310 transitions from the parity correction mode using the disk regeneration algorithm, back to the normal disk access pattern, and would not have to transition back again to the parity correction mode until service period SP 28 (not shown). Furthermore, a similar analysis is applicable for users in groups X, Y, Z, B, and C.

Paragraph [0059] beginning on page 17:

One solution to this problem comes from the design of user disk access admission algorithm employed to admit new clients onto the server. Commonly assigned U.S. patent 6,378,036, issued April 23, 2002 ~~application serial number 09/268,512, filed March 12, 1999~~, which is incorporated herein by reference, describes an information distribution system that has a queuing architecture including a plurality of queues and an associated method for scheduling disk access requests for video servers. In particular, when a new user (e.g., subscriber for services) requests admission onto the server to begin streaming a particular file (or an existing user requests a mode change), there is a demand to read a new extent off the disk that holds the beginning of the newly requested data. The server's admission policy must ascertain whether that new disk request might impinge on the guaranteed stream delivery for the server's already existing steady-state clients.

Paragraph [0061] beginning on page 18:

Furthermore, commonly assigned U.S., patent 6,691,208, issued February 10, 2004 ~~application 09/801,2001, filed March 7, 2001~~, which is herein incorporated by reference, discloses that the server's disk access model may include the stochastic nature of disk access times. For example, the time for a disk to complete a request to read 0.5 megabytes might have an average value of 33 msec, but might range from 25 msec to 80 msec, with the distribution of access times forming a Gaussian-like curve. If the server 310 has an accurate model for those access time probability distribution curves for varying sized extents, the server 310 can determine if a user's admission onto a disk 320 is allowed at a given instant, and delay the user if it is problematic because the disk 320 is too densely populated at that time. In this manner the server's admissions policy "spreads out" all the clients evenly on the disk drive array, making sure that a large number of them do not "clump up" too densely at a particular disk, as they all walk around the disk drive array 319.